S98P1097US00



(B)



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANTS: Tetsu KIMURA et al.

SERIAL NO.: 10/057,011 GROUP ART UNIT: 2171

FILING DATE: January 25, 2002 EXAMINER: Uyen T. Le

TITLE: RECORDING AND/OR REPRODUCTION APPARATUS, FILE MANAGEMENT METHOD AND PROVIDING MEDIUM

Hon. Commissioner of Patents and Trademarks,
Washington, D.C. 20231

S I R:
CERTIFIED TRANSLATION

I, Masaaki Iwami of 3-22, Asagaya-minami 1-chome, Suginami-ku, Tokyo, Japan, am an experienced translator of the Japanese language into the English language and I hereby certify that the attached comprises an accurate translation into English of Japanese Patent Application No. Hei 10-046855 filed February 27, 1998.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

December 6, 2004 — M. Iwami

Date — Masaaki IWAMI

【Document Name】 Application for Patent

【Reference No.】 9800130901

【Application Date】 February 27, 1998

【Destination】 Commissioner, Patent Office

【International Patent Classification】 G11B 7/00

【Title of the Invention】 File System for Recording and/or Reproduction Apparatus Which Uses Disc Type Recording Medium

【Number of Claims】 2

【Inventor】

【Address】 c/o Sony Corporation
7-35, Kitashinagawa 6-chome,
Shinagawa-ku, Tokyo

【Name】 Tetsu Kimura

【Inventor】

【Address】 c/o Sony Corporation
7-35, Kitashinagawa 6-chome,
Shinagawa-ku, Tokyo

【Name】 Akio Ishikawa

【Applicant for Patent】

【Identification No.】 000002185

【Name】 Sony Corporation

【Representative】 Nobuyuki Idei

【Agent】

【Identification No.】 100082131

【Patent Attorney】

【Name】 Yoshio Inamoto

【Telephone No.】 03-3369-6479

【Designation of Charge】

【Ledger No. for Prepayment】 032089

【Amount of Payment】 21,000 yen

【Domestic Priority】

【Application No.】 Hei 9-288178

【Filing Date】 October 21, 1997

【List of Filed Document】

【Object Name】 Specification 1

【Object Name】 Drawing 1

【Object Name】 Abstract 1

【Number of Comprehensive Power of Attorney】 9708842

【Necessity of Confirmation】 Necessary

[DOCUMENT NAME] SPECIFICATION

[TITLE OF THE INVENTION] File system for recording and/or reproduction apparatus which uses disc type recording medium

[CLAIMS]

[Claim 1] A file system for a recording and/or reproduction apparatus which uses a disc type recording medium, characterized in that it comprises an AV file system for recording AV data, and management information of said AV file system is recorded in an MIA (Management Information Area) and said MIA is recorded at least at two locations of a logical volume.

[Claim 2] A file system according to claim 1, characterized in that data at least of defect sectors and unused sectors are recorded in said MIA.

[DETAILED DESCRIPTION OF THE INVENTION]

[0001]

[Technical Field to which the Invention Belongs]

This invention relates to a file system to be used for a recording and/or reproduction apparatus (VDR: Video Disc Recorder) in which a disc type recording medium is used.

[0002]

[Prior Art]

As a file system for recording data onto a disc type recording medium, the ISO/IEC13346, 1995, "Information technology - Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange" is known. This file system is a general purpose file system for recording various data and is not intended to be used by an individual to record compressed digital AV (audio and vide) signals onto a disc in a home. Therefore, the system is not necessarily satisfactory to record compressed digital AV (audio and video) signals. Accordingly, a file system and a volume optimum to record AV signals are demanded.

[0003]

[Subject to be Solved by the Invention]

It is an object of the present invention to provide a file system with which an individual can record and reproduce AV signals onto and from a disc readily in a home.

[0004]

[Means to Solve the Subject]

According to the present invention, a file system for a recording and/or reproduction apparatus which uses a disc type recording medium is characterized in that it

comprises an AV file system for recording AV data, and management information of the AV file system is recorded in an MIA (Management Information Area) and the MIA is recorded at least at two locations of a logical volume.

[0005]

[Embodiment of the Invention]

2 Volume Structure

2.5.2 DescExtent

DescExtent is used to represent an area aligned with an MIB (See 2.8.2) in a descriptor recorded in the MIAs (See 2.8.2). The DescExtent must be recorded in such a form as illustrated in Table 1 below.

[0006]

[Table 1]

Desc Extent

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Offset(Number of MIB)from top of a descriptor | Uint16 |
| 2 | 2 | Length(Number of MIB) | Uint16 |

[0007]

The Offset from Top of a descriptor (RBP 0) designates an offset (the number of MIBs) from the top MIB to the area of the descriptor. The Length (RBP 2) designates a magnitude (the number of MIBs) of the area.

[0008]

2.5.3 PDLEntry

The PDL Entry (Primary Defect List Entry) is used to record a physical sector size of a physical sector for which slipping is performed in defect management. The PDL Entry must be recorded in a form indicated in Table 2 below.

[0009]

[Table 2]

PDL Entry

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Physical Sector Number of Defect Sector | Uint32 |

[0010]

The Physical Sector Number of Defect Sector (RBP 0) designates a physical sector number of a physical sector for which slipping is performed.

[0011]

2.5.4 SDLEntry

The SDL Entry (Secondary Defect List Entry) is used to record the physical sector number of a physical sector for which linear placement is performed in defect management and the physical sector number of another physical sector to be used as a substitute for the physical sector. The SDL Entry must be recorded in a form indicated in Table 3 below.

[0012]

[Table 3]

SDL Entry

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Physical Sector Number of Defect Sector | Uint32 |
| 4 | 4 | Physical Sector Number of Spare Sector | Uint32 |

[0013]

The Physical Sector Number of Defect Sector (RBP 0) designates the physical sector number of a physical sector for which linear replacement is performed.

The Physical Sector Number of Spare Sector (RBP 4) designates the physical sector number of a substitute physical sector to be used in linear replacement.

[0014]

2.6 Volume recognition

2.6.1 Physical sector

In the present paragraph, the physical sector is defined in the following manner.

- The physical sector number must be an integer which monotonously increases one by one beginning with 0.
- The physical sector size must be 2n(n > 8).

[0015]

2.6.2 Anchor points

The anchor point is a location into which information necessary to perform an analysis of the

volume structure is stored after it is discriminated that the physical volume is described in accordance with the present standards, and is a start point of a volume structure analysis. Whether the physical volume is described in accordance with the present standards or not is dependent on the discrimination that the KIFS descriptor is present in the volume recognition sequence used by the ISO/IEC13346-2/9.1.2.

The anchor descriptor is recorded in the anchor point. The physical sector number of a physical sector which makes an anchor point is not prescribed.

[Note] In a VDR, the following is prescribed by the VDR Book. In the case of a ROM (Read Only Memory) disc or a RAM (Random Access Memory) disc, Ch, 20h, LPSN-20h, LPSN-Ch are determined as anchor points. Here, LPSN stands for Last Physical Sector Number. In the case of a partial ROM disc, Ch, 20h, LPSN-20h and LPSN-Ch in ROM and RAM areas are determined as anchor points. In this instance, where appropriate information is recorded at the anchor points of the RAM area, the information is used, but where appropriate information is not recorded, the information in the ROM area is used.

An anchor descriptor is recorded in a physical sector, which serves as an anchor point, beginning with

the byte position 0. The magnitude of the anchor descriptor must be equal to or smaller than the physical sector size. Further, the area from a byte next to the last byte of the descriptor to the last byte of the physical sector is reserved for future extension, and #00 must be set to all bytes in the area. The anchor descriptor has recorded therein a definition of a main MIA area and a definition of a reserve MIA area, positions of respective MIA maps and so forth.

[0016]

2.6.3 Management Information Area (MIA)

Various pieces of information regarding a volume are recorded in a management information area (MIA). In order to secure the reliability, two MIAs having information of the same contents are recorded at different locations on the physical volume, and are individually referred to as main MIA and reserve MIA. A physical sector in each MIA is called management information block (MIB) and an offset of the physical sector number of a physical sector from the top MIB of the MIA is called management information block number (MIB number). For designation of an MIB, the MIB number of it is used. Each MIA is composed of MIBs which cannot be used because of a defect or the like, unused MIBs, and

the following MIBs used to record data.

- MIA Map for the Main MIA
- MIA Map for the Reserve MIA
- Volume Structure Descriptor
- Media Information Descriptor
- Drive Information Descriptor
- Extended Data Descriptor

For what object an MIB in an MIA is used is recorded in an MIA map. Start positions and magnitudes of the main MIA and the reserve MIA and positions of the MIA maps in the MIAs are defined by an anchor descriptor.

The data mentioned above may be recorded in a single MIB or over a plurality of MIBs. Where the data are recorded over a plurality of MIBs, in what order the MIBs are to be linked is recorded in a Map Entries field in the MIA map. Where the data come to an end intermediately of an MIB, #00 must be set to those bytes beginning with the byte next to the end of the data and ending with the last byte of the MIB.

[0017]

2.6.4 Partition

A data storage area defined by Partition Information in the Volume Structure Descriptor is called partition. One physical volume can be divided into a

plurality of partitions. A number for specifying a partition in the physical volume is called partition number. The partition number must be an integer which monotonously increases one by one beginning with 0. All of physical sectors in a partition must have an equal physical sector size.

A partition is defined as a table of partition information in the Volume Structure Descriptor. The partition information defines the partition with the physical sector number of a physical sector at the top of the partition and the number of physical sectors which belong to the partition. At least one or more partitions must be defined in the physical volume. The partition numbers are determined in an order in which the partition information is recorded in the Volume Structure Descriptor. The partition number of the partition defined by the first piece of partition information is 0, and that of the second partition is 1. In this manner, the partition number increases one by one, and that of the nth partition is n-1.

[0018]

2.6.5 Logical volume

The Logical Volume represents a data storage area defined as an aggregate of partitions in the logical

Volume Information of the Volume Structure Descriptor. The area of the Logical Volume is formed from partition areas linked in an order of description of the Partition Map of the logical volume information. The Partition Map designates partitions belonging to the logical volume with sets of the Volume Identifier which uniquely defines the physical volume and partition numbers of the physical volume. The Logical Volume may be composed of partitions which belong to different physical volumes. A single partition may belong to a plurality of Logical Volumes.

The logical volume is handled as a single area irrespective of a breakpoint between partitions or physical sectors, and contents of it are read and written in units of a logical sector. The logical sector number must be an integer which begins with 0 and monotonously increases one by one. When the magnitude of the logical volume is not a multiple of the logical sector size, a fragmental area which appears in the last physical sector is reserved for future extension and must not be used.

The Volume Structure Descriptor describes a definition of information regarding the partitions included in the physical volume, a definition of the logical volumes and so forth. In order to define a logical volume which extends over a plurality of physical

volumes, the logical volume information must be described in the Volume Structure Descriptor of that physical volume in which the partition of the partition number 0 is defined without fail.

[Note] In order to secure the reliability, the logical volume information may be described in the Volume Structure Descriptor of the physical volume to which some other partition other than the partition of the partition number 0 belongs. However the correct and the latest information must be described in the volume information descriptor of the physical volume in which the partition of the partition number 0 is defined without fail. The Volume Structure Descriptor is recorded in the MIAs.

[0019]

2.6.6 Defect management

Defect management by slipping and linear replacement can be performed for each partition. Whether or not Defect management should be performed for each partition is designated with the partition information of the Volume Structure Descriptor. A substitute data area to be used for slipping and linear replacement is called spare area. In partitions which belong to the same logical volume as that of a partition for which defect management is to be performed, one or more spare areas

must be assured without fail. Further, when linear replacement is to be performed, the last part of the partition area must be used as a spare area.

When slipping is performed, a top portion of the spare area assured at the last part of the partition area must be used as a spare area. On the other hand, where linear replacement is to be performed, for the substitute data area, a spare area other than that spare area in a partition which includes a defect sector may be used only if it belongs to the same logical volume and belongs to the same physical volume.

Information regarding slipping and linear replacement is recorded in the Defect List Information of the Volume Structure Descriptor. Information regarding slipping is recorded in the Primary Defect List while information regarding linear replacement is recorded in the Secondary Defect List.

[0020]

2.6.7 Media Information Descriptor

The Media Information Descriptor is an area into which information regarding media is to be recorded. This is used to record information regarding zones.

[0021]

2.6.8 Drive Information Descriptor

The Drive Information Descriptor is an area into which information regarding a drive (apparatus to be used for recording and reproduction of data) is to be recorded. Where a fixed drive is used, various pieces of information are recorded in the Drive Information Descriptor.

[0022]

2.6.9 Extended Data Descriptor

The Extended Data Descriptor is used to record extension information which cannot be fully recorded in the physical volume information, partition information and logical volume information.

[0023]

2.7 Volume recognition structures

2.7.1 Standard Identifier of Volume Structure Descriptor

Table 4 is added to the Standard identifier of the volume structure descriptor prescribed in the ISO/IEC13346-2/9.1.2.

[0024]

[Table 4]

Volume Structure Descriptor Interpretation

| Identifier | Interpretation |
|---|---|
| "KIFS1" | According to clause 2.7.2 |

[0025]

2.7.2 KIFS Descriptor

A KIFS descriptor must be recored in such a format as illustrated in Table 5.

[0026]

[Table 5]

KIFS Descriptor

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 1 | Structure Type | Uints = 0 |
| 1 | 5 | Standard Identifier | bytes = "KIFS1" |
| 6 | 1 | Structure Version | Uints = 1 |
| 7 | 1 | Reserved | #00 bytes |
| 8 | PSS-8 | Structure Data | #00 bytes |

[0027]

In the Structure Type (BP 0), 0 must be set.

In the Standard Identifier (BP 1), the character train ¥KIFS01" in accordance with the ISO646 must be set.

The Structure Version (BP 6) designates the version of this descriptor. 1 must be set.

The Reserved (BP 7) is reserved for future extension. #00 must be set.

The Structure data (BP 8) is reserved for future extension. #00 must be set in all bytes. PSS stands for Physical sector size.

[0028]

2.8 Volume data structures

2.8.1 Anchor Descriptor

The Anchor Descriptor must have a magnitude smaller than the physical sector size and must be recorded in a format indicated in Table 6 below.

[0029]

[Table 6]

Anchor Descriptor

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see 1.5.12) |
| 8 | 4 | Start Physical Sector Number of Main MIA | Uint32 |
| 12 | 4 | Number of Physical Sectors in Main MIA | Uint32 |
| 16 | 4 | Start Physical Sector Number of Reserve MIA | Uint32 |
| 20 | 4 | Number of Physical Sectors in Reserve MIA | Uint32 |
| 24 | 2 | Number of MIBs for MIA Map in Main MIA ($=x_1$) | Uint16 |
| 26 | 2 | Number of MIBs for MIA Map in Reserve MIA($=x_2$) | Uint16 |
| 28 | $2x_1$ | MIB Number of MIA Map for Main MIA in Main MIA | Uint16 |
| $28+2x_1$ | $2x_2$ | MIB Number of MIA Map for Reserve MIA in Main MIA | Uint16 |
| $28+2x_1+2x_2$ | $2x_1$ | MIB Number of MIA Map for Main MIA in Reserve MIA | Uint16 |
| $28+4x_1+2x_2$ | $2x_2$ | MIB Number of MIA Map for Reserve MIA in Reserve MIA | Uint16 |

[0030]

The data type field of the Signature (BP 0) must have 16 set therein.

The Start Physical Sector Number of Main MIA (BP 8) designates the physical sector number of the physical sector at the top of the main MIA.

The Number of Physical Sectors in Main MIA (BP 12) designates the number of physical sectors of the main MIA.

The Start Physical Sector Number of Reserve MIA (BP 16) designates the physical sector number of the physical sector at the top of the reserve MIA.

The Number of Physical Sectors in Reserve MIA (BP 20) designates the number of physical sectors of the

reserve MIA.

The Number of MIBs for MIA Map in Main MIA (BP 24) designates the magnitude (the number of MIBs) of the MIA map.

The number of MIBs for MIA Map in Reserve MIA (BP 26) designates the magnitude (the number of MIBs) of the MIA map of the reserve MIA.

The MIB Numbers of MIA Map for Main MIA in Main MIA (BP 28) designates MIBs in the main MIA in which the MIA map for the main MIA is recorded. The MIB numbers of the MIBs which form the MIA map must be set in order.

The MIB numbers of MIA Map for Reserve MIA in Main MIA (BP $28+2x_1$) designates the MIBs in the main MIA in which the MIA map for the reserve MIA is recorded. The MIB numbers of the MIBs which form the MIA map must be set in order.

The MIB Numbers of MIA Map for Main MIA in Reserve MIA (BP $28+2x_1+2x_2$) designates the MIBs in the reserve MIA in which the MIA map for the main MIA is recorded. The MIB Numbers of the MIBs which form the MIA map must be set in order.

The MIB Numbers of MIA Map for Reserve MIA in Reserve MIA (BP $28+4x_1+2x_2$) designates the MIBs in the reserve MIA in which the MIA map for the reserve MIA is

recorded. The MIB Numbers of the MIBs which form the MIA map must be set in order.

[0031]

2.8.2 MIA MAP

The MIA map is used to indicate a situation in use of the MIBs. The MIA map indicates the positions of MIBs which are used for recording of various data, MIBs which cannot be used because of a defect or the like, and MIBs which are not used. The MIA map must be recorded in such a form as indicated in Table 7 below.

[0032]

[Table 7]

MIA Map

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see1.5.12) |
| 8 | 2 | Location of MIA Map | Uint16 |
| 10 | 2 | Location of Volume Structure Descriptor | Uint16 |
| 12 | 2 | Location of Media Information Descriptor | Uinit16 |
| 14 | 2 | Location of Drive Information Descriptor | Uint16 |
| 16 | 2 | Location of Extended Data Descriptor | Uint16 |
| 18 | 2 | Number of Map Entries($=x_1$) | Uint16 |
| 20 | $2x_1$ | Map Entries | bytes |

[0033]

The data type field of the Signature (BP 0) must have 2 set therein.

The Location of MIA Map (BP 8) designates the MIB number of the top MIB of the MIA map.

The Location of Volume Structure Descriptor (BP 10)

designates the MIB number of the top MIB of the Volume Structure Descriptor.

The Location of Media Information Descriptor (BP 12) designates the MIB number of the top MIB of the Media Information Descriptor.

The Location of Drive Information Descriptor (BP 14) designates the MIB number of the top MIB of the Drive Information Descriptor.

The Location of Extended Data Descriptor (BP 16) designates the MIB number of the top MIB of the Extended Data Descriptor.

The Number of Map Entries (BP 18) designates the number of entries of the Map Entry beginning with BP 20. This number must be equal to the number of MIBs involved in the MIA. The value must be equal to or smaller than #FFF0.

The Map Entries (BP 20) designate the situation of use of the MIBs. One Map Entry is composed of Uint16, and the first Map Entry corresponds to the first MIB, the second Map Entry to the second MIB, …, and the nth map entry to the nth MIB. Table 8 below indicates the values of the Map Entries.

[0034]

[Table 8]

MIA Map Entry

| Value | Interpretation |
|---|---|
| #0000-#FFEF | Next MIB Number |
| #FFF0 | Unusable MIB |
| #FFF1 | Unused MIB |
| #FFF2-#FFFE | Reserved |
| #FFFF | Last MIB of the data structure |

[0035]

2.8.3 Volume Structure Descriptor

A structure of the Volume Structure Descriptor is shown in Fig. 1. Referring to FIG. 1, the @APS indicates the Align to Physical Sector, and data of the same indicates that the pertaining data must be aligned with the physical sector. Upon such alignment, #00 must be placed into an area from a byte next to the location at which data to be recorded immediately precedently is actually recorded to the end of the sector.

[0036]

2.8.4 Volume Structure Descriptor Header

The Volume Structure Descriptor Header must be recorded in accordance with Table 9 below.

[0037]

[Table 9]

Volume Structure Descriptor Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see1.5.12) |
| 8 | 2 | Descriptor Size | Uint16 |
| 10 | 2 | Reserved | #00 bytes |
| 12 | 4 | Offset to Physical Volume Information (=48) | Uint32 |
| 16 | 4 | Offset to Partition Information(=416) | Uint32 |
| 20 | 4 | Offset to Spare Area Information | Uint32 |
| 24 | 4 | Offset to Logical Volume Information | Uint32 |
| 28 | 4 | Offset to Defect List Information | Uint32 |

[0038]

In the data type field of the Signature (BP 0), 17 must be set.

The Descriptor Size (BP 8) designates the magnitude (the number of MIBs) of the Volume Structure Descriptor.

The Reserved (BP 10) is reserved for future extension and must have #00 set to all bytes thereof.

The Offset to Physical Volume Information (RPB 12) designates an offset (the number of bytes) of the Physical Volume Information from the top byte of the Volume Structure Descriptor and must have 48 set therein.

The Offset to Partition Information (RBP 16) designates an offset (the number of bytes) of the Partition Information from the top byte of the Volume Structure Descriptor, and must have 416 set therein.

The Offset to Spare Area Information (RBP 20) designates an offset (the number of bytes) of the Spare Area Information from the top byte of the Volume

Structure Descriptor.

The Offset to Logical Volume Information (RBP 24) designates an offset (the number of bytes) of the Logical Volume Information from the top byte of the Volume Structure Descriptor.

The Offset to Defect List Information (RBP 28) designates an offset (the number of bytes) of the Defect List Information from the top byte of the Volume Structure Descriptor.

[0039]

2.8.5 Physical Volume Information

The Physical Volume Information must be recorded in accordance with Table 10 below.

[0040]

[Table 10]

Physical Volume Information

| RBP | Length | Name | Contents |
| --- | --- | --- | --- |

| | | | |
|---|---|---|---|
| 0 | 2 | Character Set | Character set (see 1.5.11) |
| 2 | 2 | Physical Volume Name Size | Uint16 |
| 4 | 256 | Physical Volume Name | bytes |
| 260 | 20 | Physical Volume Identifier | bytes |
| 280 | 6 | Creation Time | Times Stamp (see 1.5.10) |
| 286 | 6 | Modification Time | Times Stamp (see 1.5.10) |
| 292 | 2 | Number of Partitions(=Np) | Uint16 |
| 294 | 2 | Number of Spare Areas(=Ns) | Uint16 |
| 296 | 2 | Number of Partitions with Defect Management(=Ndump) | Uint16 |
| 298 | 2 | Number of Logical Volume (=Nv) | Uint16 |
| 300 | 2 | Reserved | #00 bytes |
| 302 | 2 | Extended Data Identifier | Uint16 bytes |
| 304 | 64 | Extended Data | bytes |

[0041]

The Character Set (RBP 0) designates a character code of the name of a physical volume recorded in the Physical Volume Name field.

The Physical Volume Name Size (RBP 2) designates the magnitude (the number of bytes) of the name of the physical volume recorded in the Physical Volume Name field.

The Physical Volume Name (RBP 4) designates the name of the Physical Volume.

The Physical Volume Identifier (RBP 260) designates a byte train for specifying the Physical Volume uniquely for practical use.

The Creation Time (RBP 280) designates a date and time when the Volume Structure of the Physical Volume is defined for the first time.

The Modification Time (RBP 286) designates the

latest date and time when the Volume Structure of the Physical Volume is modified.

The Number of Partitions (RBP 292) designates the number of partitions included in the Physical Volume. And the value must be equal to the number of pieces of the partition information.

The Number of Spare Areas (RBP 294) designates the number of spare areas included in the Physical Volume. The value must be equal to the number of pieces of the spare area information.

The Number of Partitions with Defect Management (RBP 296) designates the number of those partitions included in the Physical Volume for which defect management is to be performed. And it must be equal to the number of the Defect List.

The Number of Logical Volumes (RBP 298) designates the number of logical volumes to which the partitions included in the Physical Volume belong. And it must be equal to the number of pieces of the logical volume information.

The Reserved (RBP 300) is reserved for future extension. It must have #00 set to all bytes therein.

The Extended Data Identifier (RBP 302) designates an ID for specifying extended data recorded in the

extended data area.

The Extended Data (RBP 304) is reserved for future extension and must have #00 set to all bytes therein.

[0042]

2.8.6 Partition Information

The Partition Information must be recorded in a form indicated in Table 11 below.

[0043]

[Table 11]

Partition Information

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Start Physical Sector Number | Uint32 |
| 4 | 4 | Number of Physical Sectors | Uint16 |
| 8 | 4 | Number of Usable Sectors | Uint16 |
| 12 | 4 | Physical Sector size(=PSS) | Uint16 |
| 16 | 1 | Access Type | Uint8 |
| 17 | 1 | Usage Information | Uint8 |
| 18 | 2 | Reserved | #00 bytes |
| 20 | 4 | Location of Primary Defect List | DescExtent (see 2.5.2) |
| 24 | 4 | Location of secondary Defect List | DescExtent (see 2.5.2) |
| 28 | 2 | Reserved | #00 bytes |
| 30 | 2 | Extended Data Identifier | Uint16 |
| 32 | 64 | Extended Data | #00 bytes |

[0044]

The Start Physical Sector Number (RBP 0) designates the physical sector number of the top physical sector in an area which forms the partition.

The Number of Physical Sectors (RBP 4) designates the number of physical sectors of the area which forms the partition.

The Number of Usable Sectors (RBP 8) designates the total number of those physical sectors in the area

forming the partition which can be used. It must be equal to the number of those physical sectors in an area except spare areas included in the area of the partition from the total area of the partition.

The Physical Sector Size (RBP 12) designates the magnitude (the number of bytes) of the physical sector of the area which forms the partition.

The Access Type (RBP 16) designates a state of a recording characteristic of the partition. Table 12 below indicates contents of the access type.

[0045]

[Table 12]

Access Type

| Value | Name | Interpretation |
|---|---|---|
| 0 | Read Only | The user may not write any data in this partition |
| 1 | Write Once | The user can write data but once in this partition |
| 2 | Rewritable | The user can write data many times in this partition |
| 3-15 | Reserved | Reserved for future use |

[0046]

The Usage Information (RBP 17) designates a state of use of the partition. Table 13 below indicates contents of the Usage Information.

[0047]

[Table 13]

Usage information

| Bit | Interpretation |
|---|---|
| 0 | Used(1:used,0:not used) |
| 1 | Defect management:Slipping(1:on,0:off) |
| 2 | Defect management:Linear replacement(1:on,0:off) |

| | |
|---|---|
| 3-7 | Reserved |

[0048]

The Reserved (RBP 18) is reserved for future extension. It must have #00 set to all bytes therein.

The Location of Primary Defect List (RBP 20) designates in the field thereof information regarding the position at which the primary defect list is recorded in accordance with 2.5.2 when defect management by slipping is to be performed for the partition. The Location of Primary Defect List (RBP 20) must have set #00 to all bytes therein when defect management by slipping is not to be performed.

The Location of Secondary Defect List (RBP 24) designates in the field thereof information regarding the position at which the secondary defect list is recorded in accordance with 2.5.2 when defect management by linear replacement is to be performed for the partition, but must have set #00 to all bytes therein when defect management by linear replacement is not to be performed.

The Reserved (RPB 28) is reserved for future extension and must have #00 set to all bytes therein.

The Extended Data Identifier (RBP 30) designates an ID for specifying extended data recorded in the Extended Data field or the extended data area.

The Extended Data (RBP 32) are reserved for future extension and must have #00 set to all bytes therein.

[0049]

2.8.7 Spare Area Information

The Spare Area Information must be recorded in a form indicated in Table 14 below.

[0050]

[Table 14]

Spare Area Information

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Start Physical Sector Number | Uint16 |
| 4 | 4 | Number of Physical Sector | Uint16 |
| 8 | 8 | Reserved | #00 bytes |

[0051]

The Stare Physical Sector Number (RBP 0) designates the physical sector number of the top physical sector of the spare area.

The Number of Physical Sector (RBP 4) designates the number of physical sectors which form the spare area.

The Reserved (RBP 8) is reserved for future extension. It must have #00 set to all bytes therein.

[0052]

2.8.8 Logical Volume Information Header

The Logical Volume Information Header must be recorded in a form indicated in Table 15 below.

[0053]

[Table 15]

Logical Volume Information Header

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Character Set | Character Set (see 1.5.11) |
| 2 | 2 | Logical Volume Name Size | Uint16 |
| 4 | 256 | Logical Volume Name | bytes |
| 260 | 2 | Boot Indicator | Uint16 |
| 262 | 2 | File System Indicator | Uint16 |
| 264 | 2 | Logical Sector Size | Uint16 |
| 266 | 2 | Number of Partitions(=Npv) | Uint16 |
| 268 | 4 | Reserved | #00 bytes |
| 272 | 16 | Logical Volume Contents Use | bytes |
| 288 | 14 | Reserved | #00 bytes |
| 302 | 2 | Extended Data Identifier | Uint16 |
| 304 | 64 | Extended Data | #00 bytes |

[0054]

The Character Set (RBP 0) designates a character code of the name of the logical volume recorded in the Logical Volume Name field.

The Logical Volume Name Size (RBP 2) designates the magnitude (the number of bytes) of the name of the logical volume designated in the Logical Volume Name Field.

The Logical Volume Name (RBP 4) designates the name of the logical volume.

The Boot Indicator (RBP 260) designates information regarding starting from the logical volume. Contents of the Boot Indicator are indicated in Table 16 below. The physical volume must not include two or more logical volumes whose boot indicators are active and whose top partitions are in the physical volume.

[0055]

[Table 16]

Boot Indicator

| Value | Name | Contents |
|---|---|---|
| 00h | Not Active | Physical volume is not set that computer boots up from this logical volume |
| 80h | Active | Physical volume is set that computer boots up from this logical volume |

[0056]

The File System Indicator (RBP 262) designates a file system used in the logical volume. Contents of the File System Indicator are indicated in Table 17 below.

[0057]

[Table 17]

File System Indicator

| Value | Name | Contents |
|---|---|---|
| 00h | Unknown | This logical volume is unknown. |
| 01h | 12bit FAT | This logical volume is formatted with 12bit FAT. |
| 04h | 16bit FAT | This logical volume is formatted with 16bit FAT. |
| 05h | 16bit FAT, Extended | This logical volume is formatted with 16bit FAT, and defined an extended partition. |
| 06h | 16bit FAT, Extended, 64KB/claster | This logical volume is formatted with 16bit FAT, and defined an extended partition, using 64KB/ claster. |
| 07h | HPFS | This logical volume is formatted with HPFS. |
| 0Bh | 32bit FAT | This logical volume is formatted with 32bit FAT. |
| F0h | KIFS | This logical volume is formatted with KIFS. |

[0058]

The Logical Sector Size (RBP 264) designates the magnitude (the number of bytes) of the logical sector of the logical volume.

The Number of Partitions (RBP 266) designates the number of partitions which form the logical volume. It must be equal to the number of the partition map.

The Reserved (RBP 268) is reserved for future extension, and must have #00 set to all bytes therein.

The Logical Volume Contents Use (RBP 272) is an area which may be used freely by the file system used by the logical volume.

The Reserved (RBP 288) is reserved for future extension and must have #00 set to all bytes therein.

The Extended Data Identifier (RBP 302) designates an ID for specifying extended data recorded in the Extended Data area.

The Extended Data (RBP 304) is reserved for future extension and must have #00 set to all bytes therein.

[0059]

2.8.9 Partition Map

The Partition Map must be recorded in a form indicated in Table 18 below.

[0060]

[Table 18]

Partition Map

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 20 | Volume Identifier | bytes |
| 20 | 2 | Partition Number | Uint16 |
| 22 | 2 | Reserved | #00 bytes |

[0061]

The Volume Identifier (RBP 0) designates a physical volume identifier recorded in the Physical Volume Information of the physical volume to which the partition which forms the logical volume belongs.

The Partition Number (RBP 20) designates the partition number of the partitions which form the logical volume.

The Reserved (RBP 22) is reserved for future extension and must have #00 set to all bytes therein.

[0062]

2.8.10 Defect List Information Header

The Defect List Information Header must be recorded in a form indicated in Table 19 below.

[0063]

[Table 19]

Defect List Information Header

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Number of MIB Primary Defect List | Uint16 |
| 2 | 2 | Number of MIB for Secondary Defect List | Uint16 |
| 4 | 12 | Reserved | #00 bytes |

[0064]

The Number of MIB for Primary Defect List (RBP 0) designates the number of MIBs used to record the primary defect list.

The Number of MIB for Secondary Defect List (RBP 2) designates the number of MIBs used to record the secondary defect list.

The Reserved is reserved for future extension and must have #00 set to all bytes therein.

[0065]

2.8.11 Primary Defect List/Secondary Defect List

The Primary Defect List/Secondary Defect List must be recorded in a form indicated in Table 20 below.

[0066]

[Table 20]

Primary Defect List/Secondary Defect List

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see 1.5.12) |
| 8 | 2 | Partition Number | Uint16 |
| 10 | 2 | Number of Entries(=Npd) | Uint16 |
| 12 | 4 | Reserved | #00 bytes |
| 16 | 4(8)Npd | Defect List Entry | bytes |

[0067]

The data type field of the Signature (BP 0) must have 18 set therein for the primary defect list, but must have 19 set therein for the secondary defect list.

The Partition Number (BP 8) designates the partition number of a partition in which the defect list is used.

The Number of Entries (BP 10) designates the number of entries of the Defect List Entry.

The Reserved (RBP 12) is reserved for future extension and must have #00 set to all bytes therein.

The Defect List Entry (RBP 16) has, for the primary defect list, the primary defect list entry recorded therein, but has, for the secondary defect list, the secondary defect list entry recorded therein. The defect list entries in both cases must be recorded in the ascending order of values of the Physical Sector Number of Defect Sector fields of the individual entries.

[0068]

2.8.12 Media Information Descriptor

The structure of the Media Information Descriptor is illustrated in FIG. 2.

[0069]

2.8.13 Media Information Descriptor Header

The Media Information Descriptor Header must be recorded in a form indicated in Table 21 below.

[0070]

[Table 21]

Media information Descriptor Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see 1.5.12) |
| 8 | 2 | Descriptor Size | Uint16 |
| 10 | 6 | Reserved | #00 bytes |
| 16 | 2 | Number of discs | Uint16 |
| 18 | 2 | Number of sides per disc | Uint16 |
| 20 | 2 | Number of layers per side | Uint16 |
| 22 | 2 | Number of zones per layer(=Nz) | Uint16 |
| 24 | 8 | Reserved | #00 bytes |
| 32 | 2 | Number of cylinders | Uint16 |
| 34 | 2 | Number of heads(tracks per cylinder) | Uint16 |
| 36 | 2 | Number of sectors per tracks | Uint16 |
| 38 | 10 | Reserved | #00 bytes |

[0071]

The data type field of the Signature (BP 0) must have 20 set therein.

The Descriptor Size (BP 8) designates the magnitude (the number of MIBs) of the Media Information Descriptor.

The Reserved (BP 10) is reserved for future extension and must have #00 set to all bytes therein.

The Number of discs (BP 16) designates the number of discs.

The Number of sides per disc (BP 18) designates the number of sides per disc.

The Number of layers per side (BP 20) designates the number of layers per side.

The Number of zones per layer (BP 22) designates the number of zones per layer.

The Reserved (BP 24) is reserved for future extension and must have #00 set to all bytes therein.

The Number of cylinders (BP 32) designates the number of cylinders.

The Number of heads (tracks per cylinder) (BP 34) designates the number of heads (number of tracks per cylinder).

The Number of sectors per tracks (BP 36) designates the number of sectors per track.

The Reserved (BP 38) is reserved for future extension and must have #00 set to all bytes therein.

[0072]

2.8.14 Zone Information

The Zone Information must be recorded in a form indicated in Table 22 below.

[0073]

[Table 22]

Zone Information

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Start Physical Sector Number | Uint16 |
| 4 | 4 | Number of Physical Sector | Uint16 |
| 8 | 8 | Reserved | #00 bytes |

[0074]

The Start Physical Sector Number (RBP 0) designates the physical sector number of the top physical sector of the zone.

The Number of Physical Sector (RBP 4) designates the number of physical sectors which compose the zone.

The Reserved (RBP 8) is reserved for future extension and must have #00 set to all bytes therein.

[0075]

2.8.15 Drive Information Descriptor

The structure of the Drive Information Descriptor is illustrated in FIG. 3.

[0076]

2.8.16 Drive Information Descriptor Header

The Drive Information Descriptor Header must be recorded in a form indicated in Table 23 below.

[0077]

[Table 23]

Drive Information Descriptor Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see 1.5.12) |
| 8 | 2 | Descriptor Size | Uint16 |
| 10 | 1 | Strategy Type | Uint8 |
| 11 | 5 | Reserved | #00 bytes |

[0078]

The data type field of the Signature (BP 0) must have 21 set therein.

The Descriptor Size (BP 8) designates the magnitude (the number of MIBs) of the Drive Information Descriptor.

The Strategy Type (BP 10) designates a Strategy Type.

The Reserved is reserved for future extension and must have #00 set to all bytes therein.

[0079]

2.8.17 Extended Data Descriptor

The structure of the Extended Data Descriptor is illustrated in FIG. 4. Referring to FIG. 4, the @APS represents Align to Physical Sector and indicates that the pertaining data must be aligned with the physical sector. Further, the area beginning with a byte next to the last data and ending with the end of the sector must have #00 set therein.

[0080]

2.8.18 Extended Data Descriptor Header

The Extended Data Descriptor Header must be recorded in a form indicated in Table 24 below.

[0081]

[Table 24]

Extended Descriptor Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature (see 1.5.12) |
| 8 | 2 | Descriptor Size | Uint16 |
| 10 | 6 | Reserved | #00 bytes |
| 16 | 2 | Location of Extended Data for Physical Volume | Desc Extent (see 2.5.2) |
| 20 | 4Np | Location of Extended Data for Partitions | Desc Extent (see 2.5.2) |
| 20+4Np | 4Nv | Location of Extended Data for Logical Volume | Desc Extent (see 2.5.2) |

[0082]

The data type field of the Signature (BP 0) must

have 22 set therein.

The Descriptor Size (BP 8) designates the magnitude (the number of MIBs) of the Extended Data Descriptor.

The Reserved (BP 10) is reserved for future extension and must have #00 set to all bytes therein.

The Location of Extended Data for Physical Volume (BP 16) designates a location in which extended data regarding the physical volume is recorded in accordance with 2.5.2.

The Location of Extended Data for Partitions (BP 20) designates a location in which extended data regarding the partitions are recorded in accordance with 2.5.2.

The Location of Extended Data for Logical Volume (BP 20+4Np) designates a location in which extended data regarding the logical volumes are recorded in accordance with 2.5.2.

[0083]

2.9 Requirements

2.9.1 Levels of medium interchange

The level 1 of medium interchange has the following limitations.

- A logical volume must be composed of partitions which belong to the same physical volume.

- Where a plurality of partitions are defined in the same physical volume, the areas of the partitions must not overlap with each other.
- All of physical sectors of the partitions which compose the logical volume must have an equal physical sector size.
- The logical sector size must be a multiple of the physical sector size, or alternatively, the physical sector size must be a multiple of the logical sector size.
- The magnitude of the partitions must be a multiple of a value of a larger one of the logical sector size or the physical sector size.
- A partition for which defect management is to be performed must assure one or more spare areas therein without fail.
- Defect management by linear replacement must use the spare area or areas assured in the partition as a substitute data area.

The level 2 has no limitation.

[0084]

2.10.1 Volume structure recognition sequence

The flow of recognition of a volume struture is such as described below.

1. The volume recognition sequence stored in the

area beginning with PSN10h is interpreted based on the ISO/IEC 13346-2/8 to recognize that this physical volume is described in accordance with the format of the KIFS.

2. The anchor descriptor is read from the anchor point to acquire information regarding the MIA.

3. The volume structure is recognized after acquiring the data of the volume structure descriptor from the MIA.

[0085]

2.10.2 Logical to physical

In order to convert the logical sector number LSN to the physical sector number, the following operation may be performed.

1. In what partition composing the logical volume the LSN belongs to is searched, then a physical sector number PSNp of the top physical sector of the partition an offset OFFp from the PSNp are found.

2. The number of physical sectors NSslip which remains unused by slipping between the top and the OFFp of the partition belonging to is found.

3. The number of physical sectors NSspare in the spare area between the top and the OFFp of the partition belonging to.

4. From the above operation, the logical sector

number LSN is converted to the physical sector number, that is,

PSNp + OFFp + NSslip + NSspare.

[0086]

2.10.3 Example of volume structure

Table 25 below illustrates a table of an example of the volume structure of a hybrid disc of FAT, ISO9660 (with Joliet), ISO/IEC13346, KIFS for a VDR. The mark ◆ represents position fixed information which cannot be re-arranged.

[0087]

[Table 25]

Example of volume structure (FAT,9660,13346,KIFS Hybrid)

| PSN(hax) | Descriptor | Contents |
|---|---|---|
| 0 | [FAT]Partition Table | ◆[FAT]Partition Table |
| - | - | - |
| c | [KIFS]Anchor Descriptor | ◆[KIFS]Anchor |
| - | - | - |
| 10 | [9660]Primary Volume Descriptor | ◆[9660/13346/KIFS] Volume recognition Sequence |
| 11 | [9660]Primary Volume Descriptor(Reserve) | |
| 12 | [9660]Supplementary Volume Descriptor(for Joliet) | |
| 13 | [9660]Volume Descriptor Set Terminator | |
| 14 | [13346]Beginning Extended Area Descriptor | |
| 15 | [13346]NSR Descriptor | |
| 16 | [13346]Terminating Extended Area Descriptor | |
| 17 | [13346]Beginning Extended Area Descriptor | |
| 18 | [KIFS]KIFS Descriptor | |
| 19 | [13346]Terminating Extended Area Descriptor | |
| - | - | - |
| 30 | [13346]Primary Volume Descriptor | [13346]Main Volume Descriptor Sequence Extent |
| 31 | [13346]Implementation Use Volume Descriptor | |
| 32 | [13346]Partition Descriptor | |
| 33 | [13346]Logical Volume Descriptor | |
| 34 | [13346]Unallocated Space Descriptor | |
| 35 | [13346]Terminating Descriptor | |
| - | - | - |
| 40 | [13346]Primary Volume Descriptor | [13346]Reserve Volume Descriptor Sequence Extent |
| 41 | [13346]Implementation Use Volume Descriptor | |
| 42 | [13346]Partition Descriptor | |
| 43 | [13346]Logical Volume Descriptor | |
| 44 | [13346]Unallocated Space Descriptor | |
| | - | - |
| 80 | [KIFS]MIA Map for Main MIA | [KIFS]Main MIA |
| 81 | [KIFS]MIA Map for Reserved MIA | |
| 82 | [KIFS]Volume Structure Descriptor | |
| 83 | [KIFS] Primary Defect List | |
| 84 | [KIFS] Secondary Defect List | |
| 85 | [KIFS]Media Information Descriptor | |
| 86 | [KIFS]Drive Information Descriptor | |
| 87 | [KIFS]Extended Data Descriptor | |
| 88 | [KIFS] Extended Data | |
| - | - | - |
| C0 | [KIFS]MIA Map for Reserve MIA | [KIFS]Reserve MIA |
| C1 | [KIFS]MIA Map for Main MIA | |
| C2 | [KIFS]Volume Structure Descriptor | |
| C3 | [KIFS] Primary Defect List | |
| C4 | [KIFS] Secondary Defect List | |
| C5 | [KIFS] Defect Sector | |
| C6 | [KIFS]Media Information Descriptor | |
| C7 | [KIFS]Drive Information Descriptor | |
| C8 | [KIFS]Extended Data Descriptor | |
| C9 | [KIFS] Extended Data | |
| - | - | - |
| 100 | [13346]Anchor Volume Descriptor Pointer | ◆[13346] Anchor |
| - | - | |
| 150 | [KIFS]LOGICAL VOLUME | |
| - | - | |
| LPSN-150 | - | - |
| - | - | - |
| LPSN-100 | [13346]Anchor Volume Descriptor Pointer | ◆[13346]Anchor |
| - | - | - |
| LPSN-20 | [13346]Anchor Descriptor | ◆[KIFS]Anchor |
| - | - | - |
| LPSN-C | [13346]Anchor Descriptor | ◆[KIFS]Anchor |
| - | - | - |
| LPSN | [13346]Anchor Volume Descriptor Pointer | ◆[13346]Anchor |

[0088]

3. AV File System

3.3.1 Logical Sector

An area into and from which data of a size of 2n (n>8) bytes can be reproduced or recorded/reproduced.

[0089]

3.3.2 Logical Sector Number

A number applied for identification of the logical sector.

[0090]

3.3.3 Logical Volume

A set composed of logical sectors which have consecutive ascending logical sector numbers beginning with 0 and have an equal size.

[0091]

3.3.4 Management Information Area (MIA)

An area composed of a plurality of successive logical sectors on the logical volume for storing various kinds of control information of the AV file system.

[0092]

3.3.5 Management Information Block (MIB)

A logical sector in the MIA.

[0093]

3.3.6 Management Information Block Number (MIB number)

A value obtained by subtracting the logical sector number of the top of the management information block of

the MIA from the logical sector number of the management information block.

[0094]

3.6 AV File System Overview

3.6.1 AV File System Descriptor

The AV File System Descriptor (see 3.8) is recorded in one logical sector. This descriptor designates the positions and the magnitudes of the main MIA and the reserve MIA on the logical volume and the positions of the MIA maps on the main MIA and the reserve MIA.

Where the system which defines the logical volume has a capability of describing attribute information of the logical volume, the position of the AV file system descriptor is designated using this function. Where the system which defines the logical volume does not have such a function, at least one AV file system descriptor must be placed at the top of the logical volume.

In order to assure the reliability, a plurality of AV file system descriptors can be recorded in the logical volume.

For the position of the AV file system descriptor where the logical volume is prescribed by the clause 2, it must be set in a form as shown in the logical volume contents Use (BP 284) field of the logical volume

information header (See 2.8.8) in Table 26.

[0095]

[Table 26]

Logical Volume Contents Use field

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Main AV File System Descriptor Location | Uint32 |
| 4 | 4 | Reserve AV File System Descriptor Location | Uint32 |
| 8 | 8 | Reserved | #00 bytes |

[0096]

The Main AV File System Descriptor Location (RBP 0) designates the logical sector number of the AV File System Descriptor.

The Reserve AV File System Descriptor Location (RBP 4) designates the logical sector number of the AV File System Descriptor which is located at a location different from that designated by the Main AV File System Descriptor Location. If only one AV file system descriptor is present on the logical volume, then #FFFFFFFF is placed in the Reserve AV File System Descriptor Location.

The Reserved (RBP 8) is reserved for future extension and must have #00 set therein.

[0097]

3.6.2 Management Information Area (MIA)

Various management information of the AV file

system is recorded in the Management Information Area (MIA). In order to secure the reliability, two MIAs having management information of the same contents are recorded at different locations on the logical volume and are individually called main MIA and reserve MIA. The positions and the magnitudes of the main MIA and the reserve MIA and the positions of the MIA maps in the MIAs are specified by the AV File System Descriptor.

A logical sector in the MIAs is called management information block (MIB), and the offset of the logical sector number of a management information block (MIB) from the top MIB of the MIA is called management information block number (MIB number). For designation of an MIB, an MIB number is used.

Each MIA is a structure of MIBs which cannot be used because of a defect or the like, unused MIBs, and MIBs for storing the following data structures:

- MIA Map
- File Table
- Allocation Extents Table
- Allocation Strategy Table
- Defect Information Table (Optional)
- Extended Attribute Table (Optional)

For what object each MIB in each MIA is used is

recorded in the MIA map (see 3.9).

The various data structures are stored in one MIB or a plurality of MIBs. Where the data structures are recorded in a plurality of MIBs, in what order the MIBs should be linked is recorded in the Map Entry field in the MIA map. If a data structure comes to an end intermediately in an MIB, then #00 must be placed in bytes of the MIB beginning with a byte next to the end of the data and ending with the last byte of the MIB.

[0098]

3.6.3 File Table

In the AV file system, files and directories are managed with a file table (See 3.10). The structure of the file table is determined depending upon the File Table Structure Type which is a parameter in the file table header.

In this document, File Table Structure Type 0 is specified (See 3.11).

In the file table structure type 0, the file table is composed of a file table header and one or more file records. A file record is a data area of a fixed length and is composed of:

- a field for identifying the file record
- a field for designating a type of the file record

- a field for designating dates and times of production and modification
- a field for designating a position and a magnitude of data
- a field for designating an attribute
- a field for designating a parent file record called Parent Link
- a field for designating a brother file record called Next Link
- a field for designating a child file record called Child Link
- a field for designating an Extended Attribute Record Chain

A number called file record number is applied to a file record, and the parent link, next link or child link is designated using such file record number.

In the File Table Structure Type 0, such a tree structure as shown in FIG. 5 wherein the first file record of the file table serves as a root is constructed. Referring to FIG. 5, each circle represents a file record, and the file record of the root is called Root File Record. Each file record which does not have data to be referred to is called directory, and each file which has data is called file. Not only a directory, but also a

file can have a child file record.

The hierarchical structure just described is implemented by setting Child Links, Next Links and Parent Links as illustrated in FIG. 6.

A list of file records composed of Next Links is called file record chain and must not include two or more records which have the same file type.

A sub file is a kind of file and indicates part of data which is referred to by a parent file record as if it were a separate file. A file record wherein the value 10 is placed in the Data Location type of the Attribute field represents a sub file.

[0099]

3.6.4 Allocation Extents Table

In the AV file system, management of data is performed in units of successive areas on a logical volume called Allocation Extent. The Allocation Extent begins with an arbitrary byte offset of a logical sector and either ends with another arbitrary byte offset in the logical sector or includes succeeding zero or more logical sector or sectors and ends with an arbitrary byte offset of another logical sector following the logical sector or sectors. The start point, end point, attribute and so forth of the Allocation Extent are recorded in the

Allocation Extent Record in the Allocation Extent Table.

The Allocation Extent Table (see 3.12) has recorded therein allocation extent records which correspond to all allocation extents on the logical volume. Each allocation extent record has a field which indicates a next allocation extent record, and a list including a plurality of allocation extent records can be produced using the field. This list is called allocation extent record chain. Normally, file data are handled as a sequenced set of allocation extents corresponding to an allocation extent record chain.

A list composed of those allocation extent records in an allocation extent table which are not used (records whose allocation extent record status (see 3.12.3) is 00) is called free allocation extent record chain and can be traced simply from the allocation extent table. Meanwhile, a list composed of those allocation extent records each of which is considered not to be recommended for re-utilization since it includes a defect in the corresponding allocation extent (those records whose allocation extent record status has the value of 10) is called defective allocation extent record chain. Also this list can be traced simply from the allocation extent table.

[0100]

3.6.5 Allocation Strategy Table

At which position an allocation extent is placed in a logical volume is determined by the Allocation Strategy. The allocation strategy table can register a plurality of allocation strategies therein and arrange an allocation extent on a logical volume using an allocation strategy different for each file. The range of an area managed by each allocation strategy or a parameter to be used by the allocation strategy table is recorded in an allocation strategy record in the allocation strategy table (see 3.13). In the file table structure type 0, an allocation strategy is determined for each file record and is recorded in the Data Location field of the file record. The Data Location field is referred to upon operation of the allocation extent so that a corresponding allocation strategy is recalled.

Two allocation strategy types of the Allocation Strategy Type 0 and the Allocation Strategy Type 1 are defined. The Allocation Strategy Type 0 is a type suitable to discontinuously handle files of comparatively small sizes such as index data while the Allocation Strategy Type 1 is a type suitable to continuously read or write data of the MPEG or the like.

[0101]

3.6.6 Defect Information Table

The Defect Information Table (see 3.14) is a table in which logical sector numbers of defect sectors in the logical volume are recorded, and can be used for management of the defect sectors. Mounting of this table is optional.

[0102]

3.6.7 Extended Attribute Table

The Extended Attribute Table (see 3.15) can be used to hold an extension attribute of a file or a directory in the MIAs.

The Extended Attribute Table is composed of an extended attribute table header and one or more extended attribute table records. An extended attribute record is a record of a fixed length having a field for a link, and an extended attribute record chain which is a list of a plurality of extended attribute records can be produced. Mounting of this table is optional.

[0103]

3.7 Signature

At the top of the data structure to be used by the AV file system, the Signature is set. The Signature must be recorded in such a manner as seen in Table 27 below.

[0104]

[Table 27]

Signature

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Identification | bytes="AVFS" |
| 4 | 1 | Version | Uint8=1 |
| 5 | 1 | Data type | Uint8 |
| 6 | 2 | Reserved | #00 bytes |

[0105]

The Identification (RBP 0) must have a character train "AVFS" set therein in accordance with the ISO/IEC 646.

The Version (RBP 4) designates a version number of this standard and must have 1 set therein.

The Data type (RBP 5) designates a type of the data structure. Depending upon the type of the data structure, one of values indicated in Table 28 below is placed in the Data type.

[0106]

[Table 28]

Data type

| Value | Interpretation |
|---|---|
| 0 | Reserved |
| 1 | AV File System Descriptor |
| 2 | MIA Map |
| 3 | File Table |
| 4 | Allocation Extents Table |
| 5 | Allocation Strategy Table |
| 6 | Defect Information Table |
| 7 | Extended Attribute Table |
| 8-255 | Reserved |

[0107]

The Reserved (RBP 6) is reserved for future extension and must have #00 set therein.

[NOTE] The Signature is presumed to be used to identify the data structure upon crush recovery.

[0108]

3.8 AV File System Descriptor

The AV File System Descriptor must be recorded in such a manner as seen in Table 29 below.

[0109]

[Table 29]

AV File System Descriptor

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 4 | Location of Main MIA | Uint32 |
| 12 | 4 | Location of Reserve MIA | Uint32 |
| 16 | 2 | Length of Main MIA | Uint16 |
| 18 | 2 | Length of Reserve MIA | Uint16 |
| 20 | 4 | Creation Time | Time Stamp |
| 24 | 4 | Modification Time | Time Stamp |
| 28 | 2 | Number of MIA Map Sectors in Main MIA ($=x_1$) | Uint16 |
| 30 | 2 | Number of MIA Map Sectors in Reserve MIA ($=x_2$) | Uint16 |
| 32 | $2x_1$ | MIA Map Sectors in Main MIA | bytes |
| $32+2x_1$ | $2x_2$ | MIA Map Sectors in Reserve MIA | bytes |

[0110]

The Signature (BP 0) is defined in 3.7. The data

type field of the Signature must have 1 set therein.

The Location of Main MIA (BP 8) designates the sector number of the start logical sector of the main MIA.

The Location of Reserve MIA (BP 12) designates the sector number of the start logical sector of the reserve MIA.

The Length of Main MIA (BP 16) designates the size of the main MIA in the form of the number of logical sectors.

The Length of Reserve MIA (BP 18) designates the size of the reserve MIA in the form of the number of logical sectors.

The Creation Time (BP 20) designates the date and the time when the AV file system descriptor is produced.

The Modification Time (BP 24) designates the date and the time when the AV file system descriptor is updated.

The Number of MIA Map Sectors in Main MIA (BP 28) designates the number of MIB numbers described in the Main MIA Map Sectors (BP 32).

The Number of MIA Map Sectors in Reserve MIA (BP 30) designates the number of MIB numbers described in the Reserve MIA Map Sectors (BP $32+2x_1$).

The MIA Map Sectors in Main MIA (BP 32) designates

MIBs which form the MIA map in the main MIA, and the MIB numbers of the MIBs which form the MIA map must be set in order in the MIA Map Sectors in Main MIA.

The MIA Map Sectors in Reserve MIA (BP $32+2x_1$) designates MIBs which form the MIA map in the reserve MIA, and the MIB numbers of the MIBs which form the MIA map must be set in order in the MIA Map Sectors in Reserve MIA.

[0111]

3.9 MIA Map

The MIA Map is used to indicate a situation of use of MIBs in the MIAs. The MIA Map indicates the positions of various data structures in the MIAs, MIBs which cannot be used because of a defect, and unused MIBs.

The MIA Map must be recorded in such a manner as seen in Table 30 below.

[0112]

[Table 30]

MIA Map

| BP | Length | Name | Contents |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 2 | Location of MIA Map | Uint16 |
| 10 | 2 | Location of Allocation Strategy Table | Uint16 |
| 12 | 2 | Location of File Table | Uint16 |
| 14 | 2 | Location of Allocation Extents Table | Uint16 |
| 16 | 2 | Location of Defect List Table | Uint16 |
| 18 | 2 | Location of Extended Attribute Descriptor | Uint16 |
| 20 | 2 | Reserved | bytes |
| 22 | 2 | Number of Map Entries(=$x_1$) | Uint16 |
| 24 | $2x_1$ | Map Entries | bytes |

[0113]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must be set to 2.

The Location of MIA Map (BP 8) designates the MIB number of the top MIB of the MIA map in the MIA.

The Location of Allocation Strategy Table (BP 10) designates the MIB number of the top MIB of the Allocation Strategy Table in the MIA.

The Location of File Table (BP 12) designates the MIB number of the top MIB of the file table in the MIA.

The Location of Allocation Extends Table (BP 14) designates the MIB number of the top MIB of the Allocation Extent Table in the MIA.

The Location of Defect List Table (BP 16) designates the MIB number of the top MIB of the Defect List Table in the MIA. If the MIA does not have a defect list table therein, then #FFFF must be placed in the Location of Defect List Table.

The Location of Extended Attribute Descriptor (BP 18) designates the MIB number of the top MIB of the

Extended Attribute Descriptor in the MIA. If the MIA does not have an extended attribute descriptor therein, then #FFFF must be placed in the Location of Extended Attribute Descriptor.

The Reserved (BP 20) is reserved for future extension and must have #00 set therein.

The Number of Map Entries (BP 22) designates the number of entries of map entries beginning with (BP 24). The number of entries must be equal to the number of MIBs existing in the MIA and equal to or smaller than #FFF0.

The Map Entries (BP 24) designate a situation of use of the MIBs in the MIA. One map entry is composed of Uint16, and the first map entry corresponds to the first MIB of the MIA, the second map entry corresponds to the second MIB, …….

The value of the map entry has such meanings as indicated in Table 31 below.

[0114]

[Table 31]

Map entry value

| Value | Interpretation |
|---|---|
| #0000-#FFEF | Next MIB Number |
| #FFF0 | Unusable MIB |

| | |
|---|---|
| #FFF1 | Unused MIB |
| #FFF2-#FFFE | Reserved |
| #FFFF | Last MIB of the data structure |

[0115]

If the data structure has a size equal to or smaller than the logical sector size and is stored in one MIB, #FFFF must be placed in the map entry corresponding to the MIB. Where the data structure is recorded over a plurality of MIBs, in the map entry corresponding to each of the MIBs other than the last MIB, the MIB number of the next MIB is placed, but in the map entry corresponding to the last MIB, #FFFF must be placed. Any MIB whose map entry has the value #FFF1 indicates that the block is unused and can be used where the data structure requires a new MIB. Any MIB whose map entry has the value #FFF0 indicates that the block is not suitable for use (because it is a defective sector or the like) (See 5.1.3)

[0116]

3.10 Generic File Table Format

3.10.1 File Table

The File Table must be composed of a file table header and file table data as seen from FIG. 7.

The structure of the file table data depends on the File Table Structure Type field of the file table header.

[0117]

3.10.2 File Table Header

The File Table Header must be recorded in such a manner as seen from Table 32 below.

[0118]

[Table 32]

File Table Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 4 | Length of File Table Data | Uint32 |
| 12 | 2 | File Table Structure Type | Uint16 |
| 14 | 18 | File Table Structure Type dependent information | bytes |

[0119]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must have 3 set therein.

The Length of File Table Data (BP 8) designates the length of the file table data in the form of the number of bytes.

The File Table Structure Type (BP 12) defines the structure of the file table data. Only 0 is defined in this document (see 3.11), 1 through 65535 are Reserved.

Information determined for each file table structure type must be set to The File Table Structure Type dependent Information (BP 14).

[0120]

3.11 File Table Structure Type 0

3.11.1 File Table

Where the File Table Structure Type is 0, the file table must be composed of a file table header and one or more file records as shown in Fig. 8.

The file records have consecutive numbers of an ascending order beginning with 0, and such numbers are called file record numbers. A list of the file records is produced by setting the file record number of a next record in the Next Link field, and is called file record chain.

All of those file records in the file table which are not used must form a file record chain called free file record chain.

[0121]

3.11.2 File Table Header

Where the file table structure type is 0, the File Table Header must be recorded in such a manner as seen in Table 33 below.

[0122]

[Table 33]

File Table Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 4 | Length of File Table Data | Uint32 |

| | | | |
|---|---|---|---|
| 12 | 2 | File Table Structure Type | Uint16(=0) |
| 14 | 2 | Number of File Records | Uint16 |
| 16 | 2 | First Free File Record | Uint16 |
| 18 | 14 | Reserved | #00 bytes |

[0123]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must have 3 set therein.

The Length of File Table Data (BP 8) must have a number obtained by multiplying the length of the file record and the Number of File Records (BP 14) together set therein.

The File Table Structure Type (BP 12) must have 0 set therein.

The Number of File Records (BP 14) designates the number of file records which compose the file table. The file record number must have a value equal to or higher than 1 but equal to or lower than #FFF0.

The First Free File Records (BP 14) designates the first element of the free file record chain and must have #FFFF set therein where the file table does not include any free file record.

The Reserved (BP 18) is reserved for future extension and must have #00 set therein.

[0124]

3.11.3 File Record

The File Record must be recorded in such a manner as indicated in Table 34 below.

[0125]

[Table 34]

File Record of File Structure Type 0

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | File ID | Uint16 |
| 2 | 2 | File Type | Uint16 |
| 4 | 4 | Attribute | Uint32 |
| 8 | 4 | Creation Time | Time Stamp |
| 12 | 4 | Modification Time | Time Stamp |
| 16 | 8 | Data Length | Uint64 |
| 24 | 8 | Data Location | bytes |
| 32 | 2 | Child Link | Uint16 |
| 34 | 2 | Next Link | Uint16 |
| 36 | 2 | Parent Link | Uint16 |
| 38 | 2 | Extended Attribute Record Number | Uint16 |

[0126]

The File ID (RBP 0) designates a number for identifying a file record having a same file type in the file record chain.

The File Type (RBP 2) designates a number for indicating the type of the file record (see 3.11.4).

The Attribute (RBP 4) designates the attribute of the file record or data to be referred to by the file record (See 3.11.5).

The Creation Time (RBP 8) designates the date and the time of production of the file record.

The Modification Time (RBP 12) designates the date and the time of modification to the file record or to the data to be referred to by the file record.

The Data Length (RBP 16) designates the length of the data to be referred to by the Data Location (RBP 24) in units of a byte. The Data Length (RBP 16) must have 0 set therein where there is no data to be referred to.

The Data Location (RBP 24) designates the position of the data to be referred to by the file record. The interpretation of the field varies depending upon contents of the Data Location Type (Bit 1-2) of the Attribute (RBP 4) (See 3.11.5).

The Child Link (RBP 32) designates the file record number of the child file record, and where no such file record is present, #FFFF must be placed in the Child Link.

The Next Link (RBP 34) designates the file record number of a next file record which composes the file record chain, and where the file record is the last element of the file record chain, #FFFF must be placed in the Next Link.

The Parent Link (RBP 36) designates the file record number of the parent file record, and where the file record is the root file record, the file record number of the file record itself, that is, 0, must be placed in the Parent Link.

The Extended Attribute Record Number (RBP 38) designates the number of the top extended attribute

record of the extended attribute record chain used by the file record, and where the extended attribute record is not referred to, #FFFF must be placed in the Extended Attribute Record Number.

[0127]

3.11.5 Attribute

The Attribute field must be recorded in such a manner as illustrated in Table 35 below.

[0128]

[Table 35]

Attribute of the File Record

| Bit | Interpretation |
|---|---|
| 0 | Valid |
| 1-2 | Data Location Type |
| 3 | Protected |
| 4 | Sorted |
| 5-31 | Reserved |

[0129]

The Valid (Bit 0) represents whether or not the file record is valid, and where it is 0, this represents that the file record is not used and the file record must be included in the free file record chain. Where the Valid is 1, this represents that the file record is used, and the file record must be reached from the root file record through child links and next links. The Data Location Type (Bit 0-1) designates the format of the Data Location (RBP 24) and has the following meanings.

- Where the Data Location Type is 00, this represents that the data location has nothing to refer to (the value 00 is set where the file record is a directory).
- Where the data location type is 01, the data location is represented in a format indicated in Table 36 below which includes the number of the top allocation extent record of the allocation extent record chain and the allocation strategy number.
- Where the Data Location Type is 10, this represents that the file record is a sub file, and the data location is represented by Uint64 as an offset of the parent file record from the top of data indicated by the data location.
- The Data Location Type 11 is reserved for future extension and must not be designated.

[0130]

[Table 36]

Data Location file of Type 01

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Reserved | #00 bytes |
| 2 | 2 | Allocation Strategy Number | Uint16 |
| 4 | 4 | First Allocation Extent Record Number | Uint32 |

[0131]

The Protected (Bit 3) represents that the file record is protected.

The Sorted (Bit 4) represents that the file record chain to which the file record belongs is sorted in an ascending order of the file type and, in the same file type, sorted in an ascending order of the file ID.

The Reserved (Bits 5-31) is reserved for future extension and must have 0 set therein.

[0132]

3.12 Allocation Extents Table

3.12.1 Structure of the Allocation Extents Table

The Structure of the Allocation Extents Table is composed of an allocation extent table header and an allocation extent record or records as seen from FIG. 9. The allocation extent records are numbered consecutively in an ascending order beginning with 0. The numbers are referred to as allocation extent record numbers. By setting the allocation extent record number of the next record to the next allocation extent record field, a list of the allocation extent records is produced. This list is referred to as allocation extent record chain.

[0133]

3.12.2 Allocation Extents Table Header

The Allocation Extents Table Header must be

recorded in such a manner as indicated in Table 37 below.

[0134]

[Table 37]

Allocation Extents Table Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 4 | Number of Allocation Extent Records | Uint32 |
| 12 | 4 | First Free Allocation Extent Record | Uint32 |
| 16 | 4 | First Defective Allocation Extent | Uint32 |
| 20 | 4 | Reserved | #00 bytes |

[0135]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must have 4 set therein.

The Number of Allocation Extent Records (BP 8) designates the number of allocation extent records in the allocation extent table.

The First Free Allocation Extent Record (BP 12) designates the first element of the free allocation extent record chain. Where the allocation extent table has no free allocation extent record, #FFFFFFFF must be placed in this field.

The First Defective Allocation Extent Record (BP 16) designates the first element of the defective allocation extent record chain. When the allocation extent table does not include a defective allocation extent record therein, #FFFFFFFF must be placed in this

field.

The Reserved (BP 20) is reserved for future extension and must have #00 set therein.

[0136]

3.12.3 Allocation Extent Record

The Allocation Extent Record represents the start position, the end position and the attribute of the allocation extent and the position of the next allocation extent record which forms the allocation extent record chain. The allocation extent record must be recorded in such a manner as indicated in Table 38 below.

[0137]

[Table 38]

Allocation Extent Record

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Start Logical Sector Number | Uint32 |
| 4 | 1 | Allocation Strategy Number | Uint8 |
| 5 | 1 | Reserved | Uint8 |
| 6 | 2 | Start Offset | Uint16 |
| 8 | 4 | End Logical Sector Number | Uint32 |
| 12 | 2 | Reserved | Uint16 |
| 14 | 2 | End offset | Uint16 |
| 16 | 4 | Attribute | Uint32 |
| 20 | 4 | Next Allocation Extent Record | Uint32 |
| 24 | 8 | Length of the Allocation Extent | Uint64 |

[0138]

The Start Logical Sector Number (RBP 0) designates a logical sector which includes the start byte of the allocation extent and must have a logical sector number placed therein.

The Allocation Strategy Number (RBP 4) designates in accordance with which allocation strategy the allocation extent record is arranged.

The Reserved (RBP 5) is reserved for extension and must have #00 set therein.

The Start Offset (RBP 6) designates a byte offset from the top byte of the logical sector including the start byte of the allocation extent to the start byte. If the start position is equal to the top byte of the logical sector, 0 is placed to the Start Offset (RBP 6).

The End Logical Sector Number (RBP 8) designates the logical sector number of the logical sector including the last byte of the allocation extent.

The Reserved (RBP 12) is reserved for extension and must have #00 set therein.

The End Offset (RBP 14) designates the offset from the top byte of the logical sector including the end byte of the allocation extent to the end byte and has 0 placed

therein if the end byte is equal to the top byte of the logical sector.

The value indicated by the Attribute (RBP 16) signifies such meaning as indicated in Table 39 below.

[0139]

[Table 39]

Attribute of the Allocation Extent Record

| Bit | Interpretation |
|---|---|
| 0-1 | Allocation Extent Record Status |
| 1-31 | Reserved |

[0140]

Where the allocation extent record status (Bit 0-1) is 01, this allocation extent record indicates a valid allocation extent and allows (possibly) normal readout. Where the bits are 11, this allocation extent record indicates a valid allocation extent, but indicates that there is the possibility that it may not allow normal readout because of presence of a defect sector or the like. Where the bits are 00, this indicates that the allocation extent record is not used at present and can be used to arrange a new allocation extent. Where the bits are 10, this indicates that the allocation extent indicated by the allocation extent record is not referred to at all, but it is not suitable to use it to arrange a new allocation extent because it includes a defect sector.

The reserved (Bit 2-31) is reserved for extension and must have 0 set therein.

The Next Allocation Extend Record (RBP 20) designates the next allocation extent record number which composes the allocation extent record chain. Where the allocation extent record is the last element of the allocation extent record chain, #FFFFFFFF must be placed in the Next Allocation Extent Record.

The Length of the Allocation Extent (RBP 24) indicates the length of the allocation extent indicated by the allocation extent record in the form of the number of bytes.

The number of bytes determined by calculation from the Start Logical Sector Number (RBP 0), Start Offset (RBP 6), End Logical Sector Number (RBP 8) and End Offset (RBP 14) must be equal to the byte number placed in the field of the Length of the Allocation Extent.

[0141]

3.13 Allocation Strategy Table

3.13.1 Structure of the Allocation Strategy Table

The allocation strategy table designates all allocation strategies used to arrange data in the logical volume by the AV file system. The allocation strategy table is composed of an allocation strategy table header

and an allocation strategy record or records as seen in FIG. 10.

[0142]

3.13.2 Allocation Strategy Table Header

The Allocation Strategy Table Header must be recorded in such a manner as seen in Table 40 below.

[0143]

[Table 40]

Allocation Strategy Table Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 2 | Number of Allocation Strategy Record | Uint16 |
| 10 | 6 | Reserved | #00 bytes |

[0144]

The Signature (BP 0) is defined in 3.7. The Signature (BP 0) must have 5 set therein.

The Number of Allocation Strategy Record (BP 8) designates the number of allocation strategy records in the allocation strategy table.

The Reserved (RBP 10) is reserved for extension and must have #00 placed therein.

[0145]

3.13.3 Generic Allocation Strategy Record Format

An allocation strategy record is used to designate an allocation strategy. The allocation strategy record must be recorded in such a manner as seen in Table 41

below.

[0146]

[Table 41]

Allocation Strategy Record

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Length of Allocation Strategy Record | Uint16 |
| 2 | 2 | Allocation Strategy Type | Uint16 |
| 4 | 1 | Allocation Strategy Number | Uint8 |
| 5 | 3 | Reserved | #00 bytes |
| 8 | $x_1$ | Allocation Strategy Type Dependent Data | bytes |

[0147]

The Length of Allocation Strategy Record (RBP 0) designates the length of the allocation strategy record in the form of the number of bytes. The length must be a multiple of 8.

The Allocation Strategy Type (RBP 2) designates the type of the allocation strategy record. (See 3.13.4, 3.13.5)

The Allocation Strategy Number (RBP 4) designates what numbered record the allocation strategy record is in the allocation strategy table. If the allocation strategy record is the first record, then 0 must be placed in the Allocation Strategy Number.

The Reserved (RBP 5) is reserved for extension and must have #00 placed therein.

The Allocation Strategy Type Dependent Data (RBP 8) has placed therein contents which are determined for each

allocation strategy type.

[0148]

3.13.4 Allocation Strategy Type 0

In this allocation strategy, the following conditions must be met.

1. The allocation extent must be arranged in an area defined by the Start Logical Sector Number (RBP 8) and the End Logical Sector Number (RBP 12) of the allocation strategy record.

2. Where part of the logical sector is allocated to a certain allocation extent, any byte of the logical sector must not belong to another allocation extent.

3. The top of the allocation extent and the top of the logical sector must coincide with each other.

The allocation strategy record of the allocation strategy type 0 must be recorded in such a manner as seen in Table 42 below.

[0149]

[Table 42]

Allocation Strategy Record of Allocation Strategy Type 0

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Length of Allocation Strategy Record | Uint16(=16) |
| 2 | 2 | Allocation Strategy Type | Uint16(=0) |
| 4 | 1 | Allocation Strategy Number | Uint8 |
| 5 | 3 | Reserved | #00 bytes |
| 8 | 4 | Start Logical Sector Number | Uint3 |
| 12 | 4 | End Logical Sector Number | Uint32 |

[0150]

The Length of Allocation Strategy Record (RBP 0) must have 16 set therein.

The Allocation Strategy Type (RBP 2) must have 0 set therein.

The Allocation Strategy Number (RBP 4) designates what numbered record the allocation strategy record is in the allocation strategy table. If the allocation strategy record is the first record, then 0 is set in the Allocation Strategy Number.

The Reserved (RBP 5) is reserved for extension and must have #00 set therein.

The Start Logical Sector Number (RBP 8) designates the top logical sector number in which the allocation extent is arranged.

The End Logical Sector Number (RBP 12) designates the last logical sector number of the area in which the allocation extent is arranged.

[0151]

3.13.5 Allocation Strategy Type 1

The allocation strategy record of the Allocation Strategy Type 1 must be recorded in such a manner as seen in Table 43 below.

[0152]

[Table 43]

Allocation Strategy Record of Allocation Strategy Type 1

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Length of Allocation Strategy Record | Uint16 |
| 2 | 2 | Allocation Strategy Type | Uint16(=1) |
| 4 | 1 | Allocation Strategy Number | Uint8 |
| 5 | 3 | Reserved | #00 bytes |
| 8 | 2 | Number of Zones(=$x_1$) | Uint16 |
| 10 | 6 | Reserved | #00 bytes |
| 16 | $16x_1$ | Zone Information Records | (see Table 49) |

[0153]

The Length of Allocation Strategy Record (RBP 0) must have the length of the allocation strategy record, that is, $16+16x_1$, set therein.

The Allocation Strategy type (RBP 2) must have 1 set therein.

The Allocation Strategy Number (RBP 4) designates what numbered record the allocation strategy record is in the allocation strategy table. If the allocation strategy record is the first record, then 0 is set in the Allocation Strategy Number.

The Reserved (RBP 5) is reserved for extension and must have #00 set therein.

The Number of Zones (RBP 8) designates the number of zone information records in the allocation strategy record.

The Reserved (RBP 10) is reserved for extension and must have #00 set therein.

The Zone Information Records (RBP 16) must have

placed therein a number of zone information records which are designated by the Number of Zones (RBP 8). The Zone Information Records must be recorded in such a manner as seen in Table 44.

[0154]

[Table 44]

Zone Information Record

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 4 | Start Logical Sector Number | Uint32 |
| 4 | 4 | End Logical Sector Number | Uint32 |
| 8 | 4 | Length of Allocation Unit | Uint32 |
| 12 | 4 | Reserved | #00 bytes |

[0155]

The Start Logical Sector Number (RBP 0) designates the start logical sector number of the zone.

The End Logical Sector Number (RBP 4) designates the last logical sector number of the zone.

The Length of Allocation Unit (RBP 8) designates an allocation unit to be arranged into the zone.

The Reserved (RBP 12) is reserved for extension and must have #00 placed therein.

[0156]

3.14 Defeft Information Table

The Defect Information Table is used to record logical sector numbers of defect sectors in the logical volume. The Defect Information Table must be recorded in

such a manner as seen in Table 45 below.

[0157]

[Table 45]

Defect Information Table

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 4 | Number of Defect Sectors(= $x_1$) | Uint32 |
| 12 | 4 | Reserved | #00 bytes |
| 16 | $4x_1$ | Defect Sector Addresses | bytes |

[0158]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must have 6 set therein.

The Number of Detect Sectors (BP 8) designates the number of entries of the defect sector address beginning with (BP 16).

The Reserved (BP 12) is reserved for extension and must have #00 set therein.

The Defect Sector Addresses (BP 16) designate logical sector numbers of defect sectors detected in the logical volume. One entry is composed of Uint32, and the values of the logical sector numbers recorded in the Defect Sector Addresses must be sorted in an ascending order.

[0159]

3.15 Extended Attribute Table

3.15.1 Structure of the Extended Attribute Table

The Extended Attribute Table is composed of an extended attribute table header and an extended attribute record or records as seen in FIG. 11.

The extended attribute records in the Extended Attribute Table are numbered consecutively in an ascending order beginning with 0, and such number is called extended attribute record number. A list of the extended attribute records is produced by placing the next record into the next extended attribute record field and is called extended attribute record chain.

Those extended attribute records which are not used in the extended attribute table must make up a list called free extended attribute record chain.

[0160]

3.15.2 Extended Attribute Table Header

The Extended Attribute Table Header must be recorded in such a manner as seen in Table 46 below.

[0161]

[Table 46]

Extended Attribute Table Header

| BP | Length | Name | Contents |
|---|---|---|---|
| 0 | 8 | Signature | Signature |
| 8 | 2 | Number of Extended Attribute Record | Uint16 |
| 10 | 2 | First Free Extended Attribute Record | Uint16 |
| 12 | 4 | Reserved | #00 bytes |

[0162]

The Signature (BP 0) is defined in 3.7. The data type field of the Signature (BP 0) must have 7 set therein.

The Number of Extended Attribute Record (BP 8) designates the number of extended attribute records in the extended attribute table and must be equal to or smaller than #FFF0.

The First Free Extended Attribute Record (BP 10) designates the first element of the free extended attribute record chain, and where the extended attribute table does not include a free extended attribute record therein, #FFFF must be set in the First Free Extended Attribute Record.

The Reserved (BP 12) is reserved for extension and must have #00 set therein.

[0163]

3.15.3 Extended Attribute Record

The Extended Attribute Record must be recorded in such a manner as seen in Table 47 below.

[0164]

[Table 47]

Extended Attribute Record

| RBP | Length | Name | Contents |
|---|---|---|---|
| 0 | 2 | Next Extended Attribute Record | Uint16 |
| 2 | 30 | Extended Attribute Information | bytes |

[0165]

The Next Extended Attribute Record (RBP 0) designates the next extended attribute record number which composes the extended attribute record chain. Where the extended attribute record is the last extended attribute record, #FFFF must have set in the Next Extended Attribute Record.

[0166]

4 PC File System

A file structure of a PC file system will be described.

A PC file system is a regular file system for a computer. There are three positions considered for a position at which the PC file system is placed.

- The PC file system is placed in a logical volume different from the one in which the AV file system is placed.
- The PC file system is placed in one of the files of the AV file system.
- A new data structure is defined in an MIA of the AV file system, the PC file system is structured in a form as united with the AV file system.

[0167]

# 5 Postscript

## 5.1 Management of Defect Sectors (informative)

### 5.1.1 General

Most of existing file systems are designed under the presumption that defect sector processing of a medium is performed in a layer positioned below the file system (for example, replacement processing in a drive). In those file systems, it cannot be discriminated where a defect sector is, and although data can be accessed at a raw transfer rate of the drive where the file system tries to access a defect-free portion of a medium, where the file system tries to access a portion of the medium for which replacement processing has been performed, data can be accessed only at a transfer rate much lower than the raw transfer rate of the drive.

Such a construction as described above has no problem with conventional computer applications because, although it is demanded to improve the average access time, estimation of individual access times is not demanded. However, in audio and video applications, if data are not supplied by a fixed amount in a fixed time, then sound or an image cannot be recorded or reproduced correctly, and therefore, it is required that a file system can perform estimation of a time required for data

accessing.

Therefore, the KIFS introduces the presumption that defect sector processing need not be performed in a lower layer and makes it possible to accurately estimate a time required for the file system to access data. Further, in the KIFS, a field or a flag for defect sector processing which has not been used in conventional file systems is prepared, and therefore, processing of a defect sector can be performed using the field or flag. Here, an example of performing defect sector processing using the function prepared by the KIFS is described.

Generally, a defect sector is detected in one of the following cases.

1. An error occurs during writing, and a defect sector is detected.

2. Although writing is completed regularly, an error is detected when the written portion is read out immediately after the writing.

3. Although writing and reading out immediately after the writing are completed regularly, an error is detected when reading out is performed after lapse of a time.

In the above cases 1 or 2, an error can be detected and dealt with upon writing by performing reading out

immediately after writing and performing an operation of confirming that the data have been written correctly. The operation is called Write and Verify.

In the case 3, an error is caused by a fault provided by dust on or damage to the optical disc. For this case, no complete countermeasure is available. However, the possibility of data loss can be reduced significantly by employing multiple writing. The KIFS principally employs the two techniques of the Write and Verify and the multiple writing to perform processing of a defect sector.

[0168]

5.1.2 Volume Structure

The volume structure is defined by:

- Volume recognition sequence
- Anchor Descriptor
- Volume Structure Descriptor
- Media Information Descriptor
- Drive Information Descriptor
- Extended Data Descriptor

Countermeasures to defect sectors for those kinds of information are taken in the following manner.

- Volume recognition sequence

This definition is given in the ISO/IEC13346 and

cannot be specified in the present standards. Since the area into which this sequence is to be stored is fixed, if a defect sector exists there and replacement processing is not performed by a drive, then the sequence cannot be recorded correctly.

However, the volume recognition sequence merely indicates based on what standards the physical volume is used, and if this information is not read normally, it can be discriminated from presence or absence of the anchor descriptor.

- Volume Structure Descriptor
- Media Information Descriptor
- Drive Information Descriptor
- Extended Data Descriptor

The volume structure descriptor, media information descriptor, drive information descriptor and extended data descriptor are managed by each MIA. The MIA can record a non-defect sector with certainty by performing write and verify without fail upon recording. Further, taking a defect, which may occur after recording, into consideration, two MIAs are recorded in an overlapping relationship at different locations, and also two MIA maps for managing a situation of use in the MIAs are recorded in an overlapping relationship at different

locations.

Further, in a volume structure volume which is defined by the volume management system, defect management by slipping or linear replacement can be performed for each of partitions which compose the logical volume.

[0169]

5.1.3 AV File System

A defect sector is dealt with in the following manner by the AV file system.

- AV File System Descriptor

When the AV file system performs writing into the AV file system descriptor, it performs Write and Verify to confirm that writing has been performed correctly. If the AV file system has failed in the writing, then it writes the AV file system descriptor at a different location and re-writes contents of the Logical Volume Contents Use field. Further, the AV file system writes two AV file system descriptors at two different locations to assure a high degree of reliability.

- MIA reliability assurance mechanism

When the AV file system performs writing into sectors in the MIAs, it performs Write and Verify to confirm that writing has been performed correctly. If the

AV file system has failed in the writing, then it writes #FFF0 into the entry field of one of the MIA maps, and then performs a same sequence of operations for a sector in the other MIA. Further, the AV file system writes the MIAs at two locations of the logical volume to assure a high degree of reliability.

- Defect Information Table

Each defect sector detected during operation by the AV file system is registered into the defect information table so that it may thereafter not be used any more.

- Defect sector in Allocation Extent

It sometimes occurs that, for data recorded in an allocation extent, a Write and Verify operation cannot be performed from a demand for the transfer rate, and only Write operation is performed. In any case, when a defect sector is detected, the AV file system determines the portion as an independent allocation extent and places 10 into the allocation extent record status of the allocation extent record of the same so that the allocation extent may be placed into the defect allocation extent record chain.

If a defect sector is detected in an allocation extent upon reading out, then the AV file system places 11 into the allocation extent record chain. When

releasing of the allocation extent is performed, the defect sector is checked, and the defect sector portion is registered as an allocation extent, whose allocation extent record status is 10, into the defect allocation extent record chain.

[0170]

[Effect of the Invention]

According to a disc format of the present invention, an individual can record and/or reproduce compressed video and audio signals simply in a home.

[BRIEF DESCRIPTION OF THE DRAWINGS]

[FIG. 1]

FIG. 1 is a view illustrating a volume structure descriptor.

[FIG. 2]

FIG. 2 is a view illustrating a media information descriptor.

[FIG. 3]

FIG. 3 is a view illustrating a drive information descriptor.

[FIG. 4]

FIG. 4 is a view illustrating an extended data descriptor.

[FIG. 5]

FIG. 5 is a diagrammatic view illustrating a file system.

[FIG. 6]

FIG. 6 is a diagrammatic view illustrating a child link, a next link and a parent link.

[FIG. 7]

FIG. 7 is a view illustrating a file table.

[FIG. 8]

FIG. 8 is a view illustrating a file table of the file table structure type 0.

[FIG. 9]

FIG. 9 is a view illustrating an allocation extent table.

[FIG. 10]

FIG. 10 is a view illustrating an allocation strategy table.

[FIG. 11]

FIG. 11 is a view illustrating an extended attribute table.

[Explanation of Reference Numerals]

None

[NAME OF THE DOCUMENT] Abstract

[ABSTRACT]

[SUBJECT] To implement a file system for allowing an individual to record and reproduce AV signals simply in a home.

[SOLVING MEANS] In a file system for a recording and/or reproduction apparatus which uses a disc type recording medium, management information of an AV file system for recording AV data is recorded in an MIA, and the MIA is recorded at least at two locations of a logical volume.

[SELECTED FIGURE] None